Modeling Relations between Objects for Referring Expression Comprehension

Ran Wensheng,[a] Tian Ran Naoaki Okazaki Kentaro Inui

Abstract: Referring Expression Comprehension (REC) is the task of pointing out the correct object in an image as corresponding to a given natural language expression. In this work, we improve a previous model of REC by explicitly aligning relations between mentions in the language expression to pairs of objects placed in specific relative positions in the image. Evaluation on the RefGoogle dataset [4] shows that our model outperforms previous work; we also find that, quite surprisingly, the image features extracted from a pre-trained convolution neural network as used by previous research are not as efficient to REC as automatically recognized category labels.

Keywords: Referring Expression Comprehension, Relation Modelling, Neural Network

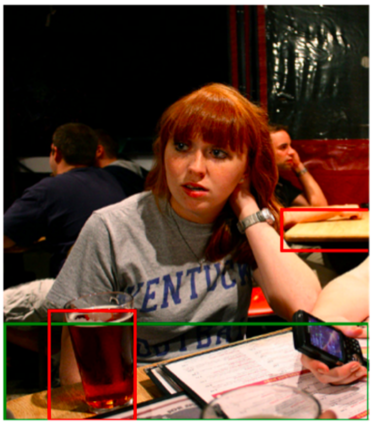


Fig. 1 Referring Expression: The desk with beer on it.

1. Introduction

We use referring expressions in our daily conversation to indicate which person or object we are pointing to, such as ”the man on the left” and ”the girl dressed in blue”. In Figure 1, there are many objects in the image, such as a beer, two desks and a woman. The expression *the desk with beer on it* can unambiguously indicate which object is being referred to. We call the phrase a *Referring Expression* indicating the object bounded by the green box.

The task of identifying a region from a given phrase is called Referring Expression Comprehension (shorted as REC) [4]. In this paper, we propose an algorithm to solve this task. Making computers understand referring expressions has applications to human-computer interaction, such as enabling robots to interact with human in the physical world. The task of REC is usually easier to evaluate compared to the image caption generation task [4], because in image caption generation, there might be more than one suitable captions to a given image; on the other hand, in REC one simply checks if the output is exactly the object referred to by the referring expression.

Resolution of referring expression requires understanding natural language and perceiving the rich visual world around us, which is a long-standing goal in the field of artificial intelligence. We have to develop models and techniques that allow us to connect the domain of visual data and the domain of natural language utterances, in order to translate between the two domains.

Referring expressions often contain information such as attribute or relation with other objects that are necessary to identify the indicated object in the image. The example in Figure 1 illustrates the necessity of relation information to resolve the task of REC: Suppose that we want to localize the object *desk* referred to by the phrase; if we do not consider the relationship between the entity *desk* and the entity *beer*, we cannot ground the referring expression since there are two desks in the image.

Previous studies either ignore the relation information between objects [4] or only model the relationship in an implicit way [6]. In these approaches, the referring expression is embedded as a vector which is generated from a language model conditioned on an image representation. Even if more than one entities are mentioned in the phrase, their relationship is not explicitly modeled since the phrase is represented by a single vector.

In this work, we explicitly incorporate relation information by mapping relations between entities in the phrase to the spatial relation between the corresponding objects in the image. More specifically, we first extract entities from the referring expression and objects from the image respectively, then learn an alignment between entities and objects. In this alignment, relations between entities and relative positions of objects are explicitly considered.

[1] Information Processing Society of Japan, Chiyoda, Tokyo 101–0062, Japan
[2] Johoshori University, Chiyoda, Tokyo 101–0062, Japan
[†1] Presently with Johoshori University
[a] 6657291qq@gmail.com

For example, in Figure 1 we can extract two entities, *the desk* and *beer* from the phrase and three objects from the image. The relation ”*with*” between the two entities is paired with the positions of any two objects in the image to calculate a score, which models the appropriateness of the alignment. We evaluate our model on Google Refexp dataset[4] and show that our result outperforms the baseline method proposed in [4].

2. Related Work

REC is a classic Natural Language Processing(NLP) problem [3]. Before the deep learning methods are widely used in this field, most works focused on a relatively small dataset of artificial objects and the text comprehension module and vision module are separated. They have to explicitly enumerate all attributes (size, color. etc) or relationships predefined in order to understand the referring expression thus can’t flexibly deal with abundant natural expressions in real world. [4] is the first one to apply deep learning methods to referring expression comprehension and generation. Their contributions are twofolds. First they released a large size dataset called Google Refexp which is used in the experiment of this paper. Their second contribution is proposing an CNN-LSTM framework for both comprehension and generation problems. They use CNN (Convolutional Neural Network) to extract information from candidates region in the form of vector and feed that vector as input to LSTM language model. And the CNN features for each candidate object is not only extracted from the object’s region but also from the whole image served as a context. In addition, the axis information of objects’ region (also known as bounding box) is incorporated into the visual representation of object along with CNN features. They try to learn the parameter of the model by letting the probability of referring expression generated from the target object larger than that generated from other objects.

The limitation of [4] is that it doesn’t consider the relationship of target object with other objects in the image. Especially when there are multiple instances of the same type presented in the image, it is often not enough to distinguish the target object from other objects with same type by encoding only the attributes information of the object itself.

Towards that end, [6] and [1] tries to encode the relation information to the visual representation of the object. More specifically, in [6], the input CNN features are obtained from a {*region*,*context_region*} pair where the whole image together with other objects in the image are considered as one of the context region. They use LSTM [7] to calculate the probability of a referring expression for one object. As the bounding boxes for context objects are not available during training, they use Multiple-Instance Learning(MIL) objective function. In [1]’s work, they take a more focused approach to encode context information. They add object comparison features to the visual representation besides CNN features for the object region and the context region. Object comparison features are the difference vector between target object’s CNN feature and other objects’ which are also presented in the image. Even though they make an effort to incorporate the relation information into the model, it is performed in an implicit way. [24] models the relationship explicitly by parsing the referring expression using recurrent network with attention. Their limitation is that the phrase is forced to be parsed into three parts called *subject, relationship, object* even though there may be more than two entities mentioned in the phrase.

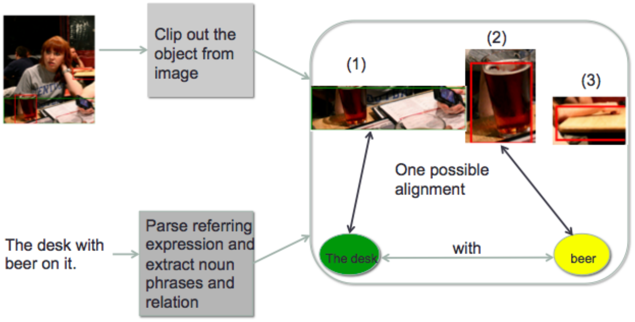

Clip out the object from image
(1)
(2)
(3)
One possible alignment
The desk with beer on it.
Parse referring expression and extract noun phrases and relation
The desk
with
beer

Fig. 2 An illustration of the proposed method.

All approaches mentioned above tries to ground the whole phrase to a region or several regions in the image. There are also many work that do the grounding in a fine-grained style. It is not the whole phrase but the entity mentions in the phrase that are grounded to the image. [20] use dependency parsing to divide phrases into several fragments and learn the alignment between regions in the image and fragments. [21] utilizes bidirectional Recurrent Network to model phrases instead of dependency parsing and align language and image modalities through a multimodal embedding. Both works are evaluated on image retrieval task. [19] learns grounding by reconstructing a given phrase using an attention mechanism. The phrase is first encoded by LSTM language model and then the embedding is decoded back to phrase together with attentions over visual representations of all objects in the image. [22] models relationship between objects presented in the image by training visual models for objects and language models for predicates individually and later combines them together to predict multiple relationships per image. [23] also proposed a model to capture the relationship between objects by doing an exhausted investigation on spatial and visual features.

3. Model

In this section, we will introduce the proposed method in detail. Given an image with several objects presented in it, and a referring expression pointing to the target object, we first parse the referring expression to extract all noun phrases (noted as NP) and the relationships between them. Then an alignment between the referring expression and the image is constructed as a bipartite, in which noun phrases in the referring expression and objects in the image are considered as nodes, and a correspondence between them is represented as a configuration of edges in the bipartite graph. Given the nodes, we assign a score to every possible graph to measure how well the alignment is. We use machine learning methods to learn parameters of the scoring function.

3.1 Constructing Bipartite

For each input pair of referring expression and candidate objects, we construct a graph and assign a scalar score Score(x, y) to the bipartite graph. We want the model to learn parameters automatically so that it can assign high score to the correct graph and low score to wrong graphs. A correct graph means that all the noun phrase nodes are correctly connected to the object nodes they refer to. Since there are many possible graphs given an input pair, our task is to search for a graph that has the maximum score.

$$\hat{y} = argmax_{y \in G(x)} \mathrm{Score}(x, y) \qquad (1)$$

Here $G(x)$ is the set of all possible graphs given an input pair x of referring expression and candidate objects. For example in Figure 2, x is composed of 3 nodes of object regions on one side of the graph and 2 nodes of noun phrase on the other side of the graph. $\hat{y}$ stands for the correct graph and y stands for an arbitrary graph in $G(x)$. Score(x, y) is composed of local score and global score. More specifically, it is defined as:

$$\mathrm{Score}(x, y) = \sum_{e \in E(y)} score_l(x, e) + \sum_{e_i, e_j \in E_{pair}(y)} score_g(x, e_i, e_j) \qquad (2)$$

$$score_l(x, e) = \boldsymbol{\theta_l} \cdot \boldsymbol{\phi_l}(x, e) \qquad (3)$$

$$score_g(x, e_i, e_j) = \boldsymbol{\theta_g} \cdot \boldsymbol{\phi_g}(x, e_i, e_j) \qquad (4)$$

where $E(y)$ is the set of edges in graph y, and $E_{pair}(y)$ is the set of pairs of edges whose noun phrase nodes are in a relationship. $score_l$ stands for local score and captures how well the entity mentioned in referring expression is matched with object in the image. $score_g$ stands for global score, which measures the fitness of the textual representation between entities in referring expression and the spatial relationship of objects in the image. θ_l and θ_g are the parameters of the model to be learned and $\boldsymbol{\phi_l}(x, e)$ and $\boldsymbol{\phi_g}(x, e_i, e_j)$ represent local features and global features. For example, in figure 1, the score of the graph will be the sum of local score *region 1, the desk* and *region2, beer*, together with global score *with, region1, region2*.

The score for an edge (local score) or a pair of edges (global score) are calculated using the dot product of a feature vector (local feature or global feature) with model parameters. To compute the local score of an edge, we first extract representations for both entity node from noun phrase and object node from bounding box in the image. More specifically, the representation of entity node , noted as w_i, is calculated by averaging the embeddings of words in noun phrase. In our setting, the word embeddings are initialized with pre-trained word2vec [16],which is a 300-dimensional vector. And the representation for object node, noted as v_i, includes the visual feature extracted from the bounding box of object in the image using pre-trained convolutional neural network model VGG-16 [13]. We also incorporate spatial information of the bounding box,noted as s_i, to object representation. $s_i = [\frac{s_{x,min}}{W_I}, \frac{s_{y,min}}{H_I}, \frac{s_{x,max}}{W_I}, \frac{s_{y,max}}{H_I}, \frac{S_{region}}{S_{image}}]$, where W_I and H_I are the width and height of the image from which the candidate object comes, and $s_{x,max}, s_{x,min}, s_{y,max}, s_{y,min}$ are the top left and bottom right coordinates of the bounding box for candidate object. S_{image} is the area of the whole image and S_{region} is the area of the region. This finally results in a 1005-dimensional vector $v_{s,i} = [v_i, s_i]$.

Once we get the representation for both object node and entity node, we calculate the local feature that incorporates both textual and visual information. Since the dimensions for $v_{s,i}$ and w_i are different, we need to perform linear transformation on $v_{s,i}$ to make the dimension of two vectors the same. Here we adopt the element-multiplication method to combine representations from language and vision since it has been shown to be a powerful way to combine representations from different modalities [2]. So we have the following formulas.

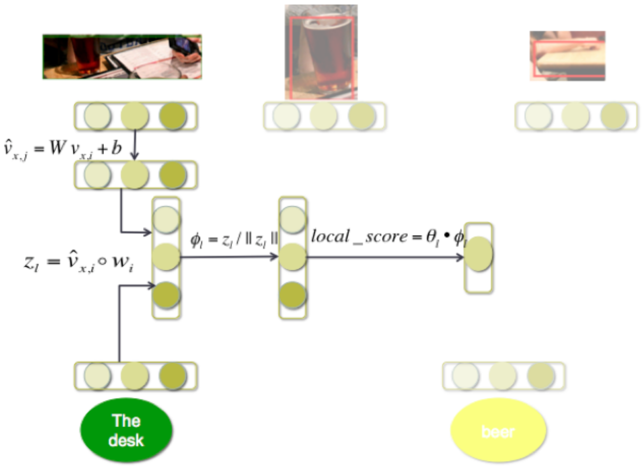

$\hat{v}_{x,j} = W v_{x,i} + b$
$z_l = \hat{v}_{x,i} \circ w_i$
$\phi_l = z_l / \| z_l \|$
$local\_score = \theta_l \bullet \phi_l$
The desk
beer

Fig. 3 The figure gives an example of the calculation of the local score for an edge. θ_l, W, b are model parameters, w_i is the average of word vectors in noun phrase, $v_{x,i}$ stands for the visual representation of object in the image.

$$\hat{v_{s,i}} = W_{x,i} v_{s,i} + b_{x,i} \qquad (5)$$

$$z_{loc} = \hat{v_{s,i}} \odot w_i \qquad (6)$$

$$\boldsymbol{\phi_l}(x, e) = \hat{z_{loc}} = z_{loc}/||z_{loc}|| \qquad (7)$$

where $\odot$ is the element-wise multiplication between two vectors. Now that the local feature $\boldsymbol{\phi_l}(x, e)$ is available from Equation(7), we can use it to calculate the local score with Equation (3). Figure 3 illustrates the mechanism of local score computation.

For global scores, we calculate the global feature $\boldsymbol{\phi_g}(x, e_i, e_j)$ by using the spatial information of two bounding boxes, noted as $s = [s_i, s_j]$, and the average embeddings of words in relationship phrase, noted as w. More concretely, we have:

$$\hat{s} = Ws + b \qquad (8)$$

$$z_{rel} = \hat{s} \odot w \qquad (9)$$

$$\boldsymbol{\phi_g}(x, e_i, e_j) = \hat{z_{rel}} = z_{rel}/||z_{rel}|| \qquad (10)$$

Finally, the global score are calculated from global feature and model parameter with Equation (4). Figure 4 illustrate the mechanism of global score computation.

3.2 Learning

In our proposed model, parameters to be learned come from the embedding of noun phrases and their relationships, along with the weight matrices of local score and global score. Our objective is to minimize the following function:

$$J(\boldsymbol{\theta}) = \min \sum_{k=1}^{N} l_k(\boldsymbol{\theta}) \qquad (11)$$

where

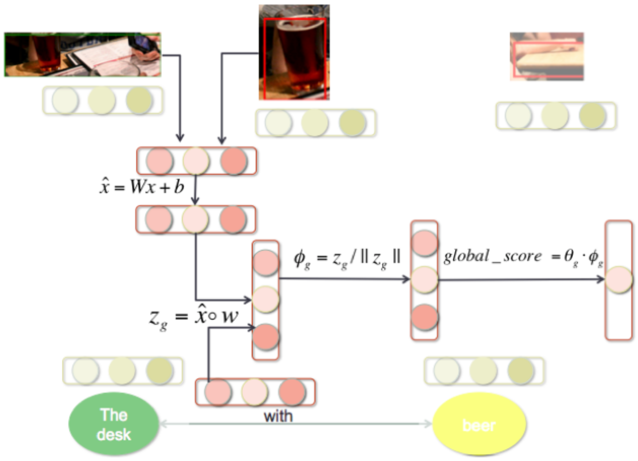

$\hat{x} = Wx + b$
$\phi_g = z_g / \| z_g \|$
$global\_score = \theta_g \cdot \phi_g$
$z_g = \hat{x} \circ w$
The desk
with
beer

Fig. 4 The figure gives an example of the calculation of the global score for an edge. θ_g, W, b are model parameters, wi is the average of word vectors in relation between noun phrases, x stands for the spatial informations for the bounding boxes of two objects.

$$l_k(\boldsymbol{\theta}) = \max_{g_k \in G(x)} \Big(Score(x_k, g_k; \boldsymbol{\theta}) - Score(x_k, \hat{g}_k; \boldsymbol{\theta}) + \|g_k - \hat{g}_k\|_1 \Big) \qquad (12)$$

Note that $\hat{g}_k$ represents the correct graph and g_k is the candidate graph. $\|g_k - \hat{g}_k\|_1$ denotes the Hamming distance between the candidate graph and the correct graph. N is the total number of instances in the training set and $l_k(\boldsymbol{\theta})$ is the loss for the k-th instance. The meaning of this formula is that we want the score for correct graph to be larger than the candidate graph by a certain margin. Once we learned the optimized parameter for all the training data, then for new data, we just need to calculate the score for all possible graphs and then choose the graph with the largest score. Then under the graph with largest score, we output the object that is connected to the entity being referring to in the referring expression.

In the problem setting, we only know which object is being referred to by the referring expression. That means the correspondence between the entity that served as the subject part in referring expression (note as subject entity) and the target object in the image is known, but we do not know the entity that has relationship with the subject entity points to which object in the image. For example, in figure 2. We know the entity *The desk* and *Region 1* is matched with each other, but we do not know entity *beer* and *Region 2* are matched in advance. We expect the model to learn these latent alignments automatically.

Specifically, we first take a non-subject entity and align it to an object. The edge is sampled with a probability proportional to the local score between the non-subject entity and the object calculated by the current model. When all the non-subject entities are aligned to their objects, we note the graph that matches the subject entity with correct object as A, and the graphs that match the subject entity to other objects as B, and and modify the parameters of the model so that the score A is larger than scores for B.

4. Experiment

4.1 Dataset

The proposed method is evaluated on Google Refexp [4] which is constructed on top of the MSCOCO [5] dataset, a dataset composed of images of complex everyday scenes containing common objects in their natural context. They selected images from the MSCOCO dataset that contain at least 2 instances of the same object type and the bounding box of the object in the image occupy at least 5% of the image. Then they constructed a Amazon Mechanical Turk task in which they present each object in the image and let the worker to generate a unique text description of that object. They also constructed a second task in which a different worker is asked to click the object given the referring expression generated in the first task. If the clicked object overlaps with the true object, then the referring expression are considered valid and added to the Google Refexp dataset. This results in a dataset of 54822 objects in 26711 images and 104560 expressions. The dataset are split to a validation set with 5000 objects, a test set with 5000 objects and a train set with remaining objects. Since the author of Google Refexp dataset only published train set and validation set, we report the accuracy of the models on the validation set.

4.2 Evaluation Metrics

In this section, we describe how to evaluate the performance of the proposed model. We compare the Intersection over Union (IoU) ratio between the bounding box of the true object and the the bounding box of the object predicted by the model. If IoU is larger than 0.5, we consider the output of the model to be true. We compute the percentage of true prediction objects over the validation set.

4.3 Implementation Details

We use Stanford CoreNLP [15] to parse referring expressions. We perform both constituency parsing and dependency parsing on input phrases to extract all noun phrases and relation expressions between them. We initialize the embeddings of all the words in noun phrases and relation expressions with word2vec [16], which results in a 300-dimensional word representations. For visual representation, we extract CNN features from bounding box region of the object using the 16-layer VGGNet [13] pretrained on the ImageNet[12]. We use the 1000 dimensional vectors from the last layer(fc8) of VGGNet and fine tune only the last layer while keeping everything else fixed. The visual feature for object is the CNN feature concatenated with the spatial information of the bounding box explained in section 3.

We implement our program using Chainer [18]. We use Adam[17] with a learning rate of 0.01. The batchsize is 16. We do not use any regularization method.

4.4 Results

4.4.1 Result on the Validation Data

We implement the method proposed in [4] and compare the accuracy on validation data of their method with our proposed method. And in order to check whether the global score component is necessary for this task, we remove the global score component and experiment the remain model on the validation data. The results are shown on Table 1. It can be seen from the result that the accuracy of our reimplementation almost reached

Table 1 Performance of the baseline and proposed method. The first line is the accuracy reported on [4]. The second line is the accuracy of our reimplementation. The third line is the accuracy of our proposed model and the fourth line is the accuracy of proposed model without global score.

Methods	Accuracy
Mao+,2015	42.50%
Mao+,2015 My implementation	40.01%
Proposed Method	44.01%
Proposed Method (no global score)	34.67%

the result of baseline method proposed in [4] and our proposed algorithm outperforms the baseline model. The fourth line of Table 1 shows the result of our proposed algorithm with the component of global score. The extremely decreasing accuracy indicates that the global score is a necessary component of our model.

4.4.2 Result on the Subset of Validation Data

In order to explicitly illustrate the proposed method's ability of modeling relationships between noun phrases in referring expression, we randomly sample 200 instances from both training set and test set and pick out instances manually that requires relationships with other objects in order to get the right answer. This results in 37 instances for train set(called set A), and 42 instances for validation set(called set B), we apply both baseline method [4] and our proposed method to these two datasets. We also remove the global score part from our proposed model and check how the accuracy changes.

The results are shown in Table2. From table 2, we observe that improvement on Set A and B from [Mao+,2015] to proposed method (around 19%) is larger than that on full validation set (about 4%). The decrease on Set A and B is also larger than that on the full validation set. Furthermore, the first line of Table 2 shows that accuracy on Set B (32.69%) is lower than that (40.10%) on the full validation set, which indicates that the baseline model is not good at dealing with instances that requires relation information. On the other hand, our proposed method achieves better accuracy on Set B (50.0%) compared to the full validation set (44.01%) which strongly shows our proposed method's capability of processing relationship information superior to the baseline method in [4].

4.4.3 Instance Analysis

Figure 5 shows some positive examples from Google Refexp [4]. There are three columns; the left column stands for the input, the middle column is the output of the baseline model and the right column is the output of our proposed method. For the first example in the first row, the proposed method can choose the correct object by matching the noun phrase *wooden chair* to green box and noun phrase *lady* to the red box. On the other hand, the baseline model failed to choose the correct object. For the second example in the second row, the proposed method points out the correct object by matching the noun phrase *A giraffe* to the green bounding box in the image and noun phrase *a zebra* to the red bounding box, while the baseline model failed to capture the meaning of the referring expression and chooses the wrong object *zebra*.

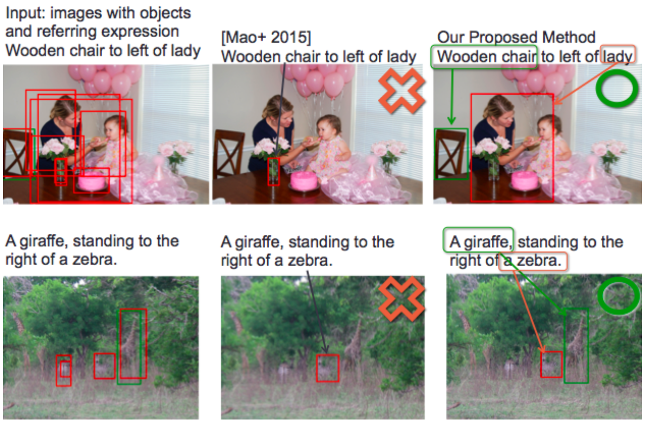
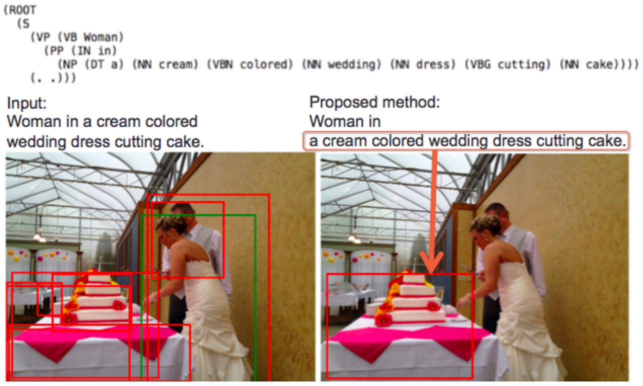

Input: images with objects
and referring expression
Wooden chair to left of lady
[Mao+ 2015]
Wooden chair to left of lady
Our Proposed Method
Wooden chair to left of lady
A giraffe, standing to the
right of a zebra.
A giraffe, standing to the
right of a zebra.
A giraffe, standing to the
right of a zebra.

Fig. 5 Example of instance for baseline method and proposed method.



```
(ROOT
  (S
    (VP (VB Woman)
      (PP (IN in)
        (NP (DT a) (NN cream) (VBN colored) (NN wedding) (NN dress) (VBG cutting) (NN cake))))
    (. .)))
```

Input:
Woman in a cream colored
wedding dress cutting cake.
Proposed method:
Woman in
a cream colored wedding dress cutting cake.

Fig. 6 Negative example of our proposed method.

There are also some negative examples which shows the limitation of our proposed model. For example in Figure 6, the Stanford CoreNLP gives the wrong parsing result by parsing the noun ”woman” into a verb. As a result, only noun phrase ”cake” is extracted from the referring expression and leads to the wrong object in the image.

Other negative example is shown Figure 7. Notice that both baseline method and our proposed method failed to match the entity ”a human hand” to the right object in the image. And both methods match the entity to a totally different object (a pillow and a hamburger) which has completely no connection with ”human” or ”hand” . For all experiments until now we use features extracted from VGG model[13] as the visual representation for object in the image and the visual feature is considered to contain abundant information about the object. But according to the negative example in Figure 7, even the fundamental information such as the category of the object is not restored in the visual feature. We assume that visual feature extracted from VGG model is not suitable for this task and experiment on another visual representation of objects.

4.5 CNN feature or category information as visual representation of image?

Since the visual features extracted from pre-trained CNN model for candidate objects do not perform very well, we also experiment with the category information of the objects. Because all the annotated objects in the dataset can be categorized into

Table 2 Comparison of the baseline and proposed method on subset.

Methods	Accuracy on Set A	Accuracy on Set B	Accuracy on Validation Set
reimplementation of Mao+,2015	67.80%	32.69%	40.10%
Proposed Method	86.48%	50.0%	44.01%
Proposed Method(no global score)	40.47%	27.02%	34.67%

Table 3 Comparison of different visual representation on validation set.

Accuracy on Test Data	[Mao+,2015]	Proposed Method
CNN vector as visual representation	40.10%	44.01%
Category information as visual representation	57.69%	61.77%

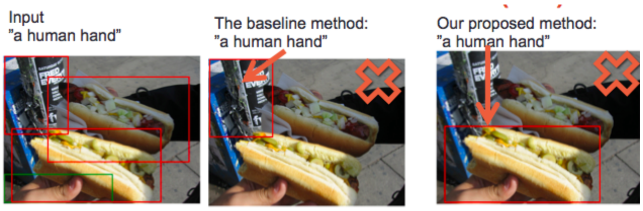


Fig. 7 Other negative example of our proposed method.

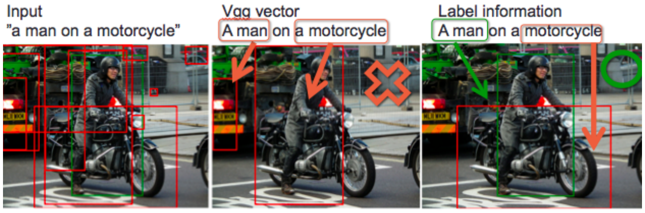


Fig. 8 Comparison of vgg vector and label information.

Table 4 Label and words that has high similarity with it.

Label	The words that have high similarity with label
carrot	leafy, sweet, carrot
donut	donut
sandwich	eggs, hamburger, sandwich
oven	stove, gas, oven, candle, burnt
airplane	seating, suitcase, tourist, china
couch	pillows, furniture, chair, man, woman, sofa
TV	monitor, television, computer
bowl	pans, buffet, fries, cup, container
clock	time, digital
bottle	squirting, beverage
vase	flower, vase

Table 5 Result of classifier over 80 categories

Accuracy on Train Set	Accuracy on Test Set
38.4%	33.4%

80 classes, we use one-hot vector with length of 80 as the visual representation for the candidate object instead of features extracted from CNN. The results are shown on Table 3. We observe a significant improvement by adopting the category information on both baseline method and our proposed method. This illustrates that category informations are more effective to represent the visual content of the object candidates and the reason may be that the CNN model pre-trained on [12] are not perfectly matched with the Google Refexp dataset.

An example is shown in Figure 8. In Figure 8, the second column shows the output of our proposed method with VGG vector as visual representation and the third column shows the output of our proposed method with one-hot category vector as visual representation. We can see that in VGG vector setting, the entity *a man* and *a motorcycle* are matched to wrong objects while in the one-hot category vector setting, both entities are matched to the correct objects.

To confirm the assumption that the visual feature extracted from CNN does not contain the category information, we try to learn a classifier for 80 categories in our dataset from VGG-16[13] vector extracted from object in the image. There are totally 210775 objects in the Google Refexp dataset, noted as red bounding box in the image of all example instances. We take 200775 instances as train set and the other 10000 instance as test set. A two-layer fully connected neural network is adopted as the classifier. The result illustrated in Table 5 shows that accuracies on both train set and test set are relatively low which verifies our assumption that the category information are not included in the VGG vectors.

Finally, we compare the representation of category labels learned by the model with word vectors in order to explore the semantic meaning captured by the label representation. The results are shown in Table 4. We check category vector learned by our model. More specifically, for each category in Google Refexp, the category vector is brought to the formula (6) and represent $\hat{v}_{x,i}$, we calculate the local score with all words in the vocabulary by replacing the w_i in formula (6) with the word embedding (according to formula (7)), and sort the word by the local score. We list up 20 words according to the score and show the result in Table 4.

5. Conclusion and Future Work

In this paper, we proposed a graph-based method that outperforms the baseline method by analyzing the syntax structure of the referring expression for the object in the image. We try to construct the system that can map the noun phrases in expression to the candidate objects in the image in order to explicitly get the context object mentioned in the expression. And we also check whether the VGG vector is suitable for the visual representation of objects in the image. We plan to do more analysis in future work.

Acknowledgments This work was supported by JST CREST Grant Number JPMJCR1513, Japan.

References

[1] Licheng Yu and Patrick Poirson and Shan Yang and Alexander C. Berg and Tamara L. Berg: Modeling Context in Referring Expressions ECCV2016

[2] Jimmy Ba and Volodymyr Mnih and Koray Kavukcuoglu: Multiple Object Recognition with Visual Attention ICLR2015

[3] Krahmer, Emiel and van Deemter, Kees: Computational Generation of Referring Expressions: A Survey Comput. Linguist. 2012

[4] Mao, Junhua and Huang Jonathan and Toshev, Alexander and Camburu, Oana and Yuille, Alan and Murphy, Kevin Generation and Com-

prehension of Unambiguous Object Descriptions CVPR2016

[5] Tsung-Yi Lin and Michael Maire and Serge J. Belongie and Lubomir D. Bourdev and Ross B. Girshick and James Hays and Pietro Perona and Deva Ramanan and Piotr Dollár and C. Lawrence Zitnick Microsoft COCO: Common Objects in Context http://arxiv.org/abs/1405.0312

[6] Varun K. Nagaraja and Vlad I. Morariu and Larry S. Davis Modeling Context Between Objects for Referring Expression Understanding ECCV2016

[7] Hochreiter, Sepp and Schmidhuber, Jürgen Long Short-Term Memory Neural Comput. November 15, 1997

[8] Ilya Sutskever and Oriol Vinyals and Quoc V. Le Sequence to Sequence Learning with Neural Networks NIPS 2015

[9] Yonghui Wu and Mike Schuster and Zhifeng Chen and Quoc V. Le and Mohammad Norouzi and Wolfgang Macherey and Maxim Krikun and Yuan Cao and Qin Gao and Klaus Macherey and Jeff Klingner and Apurva Shah and Melvin Johnson and Xiaobing Liu and Lukasz Kaiser and Stephan Gouws and Yoshikiyo Kato and Taku Kudo and Hideto Kazawa and Keith Stevens and George Kurian and Nishant Patil and Wei Wang and Cliff Young and Jason Smith and Jason Riesa and Alex Rudnick and Oriol Vinyals and Greg Corrado and Macduff Hughes and Jeffrey Dean Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation http://arxiv.org/abs/1609.08144

[10] Kelvin Xu and Jimmy Ba and Ryan Kiros and Kyunghyun Cho and Aaron C. Courville and Ruslan Salakhutdinov and Richard S. Zemel and Yoshua Bengio Show, Attend and Tell: Neural Image Caption Generation with Visual Attention ICML2015

[11] Kaiming He and Xiangyu Zhang and Shaoqing Ren and Jian Sun Deep Residual Learning for Image Recognition http://arxiv.org/abs/1512.03385

[12] Olga Russakovsky and Jia Deng and Hao Su and Jonathan Krause and Sanjeev Satheesh and Sean Ma and Zhiheng Huang and Andrej Karpathy and Aditya Khosla and Michael S. Bernstein and Alexander C. Berg and Fei-Fei Li ImageNet Large Scale Visual Recognition Challenge http://arxiv.org/abs/1409.0575

[13] Karen Simonyan and Andrew Zisserman Very Deep Convolutional Networks for Large-Scale Image Recognition http://arxiv.org/abs/1409.1556

[14] Alex Krizhevsky and Sutskever, Ilya and Hinton, Geoffrey E ImageNet Classification with Deep Convolutional Neural Networks NIPS2012

[15] Manning, Christopher D. and Surdeanu, Mihai and Bauer, John and Finkel, Jenny and Bethard, Steven J. and McClosky, David The Stanford CoreNLP Natural Language Processing Toolkit http://www.aclweb.org/anthology/P/P14/P14-5010

[16] Tomas Mikolov and Kai Chen and Greg Corrado and Jeffrey Dean Efficient Estimation of Word Representations in Vector Space http://arxiv.org/abs/1301.3781

[17] Diederik P. Kingma and Jimmy Ba Adam: A Method for Stochastic Optimization http://arxiv.org/abs/1412.6980

[18] S.TOkui K.Oono S.Hido and J.Clayton Chainer: a next-generation open source framework for deep learning NIPS Workshop on Machine Learning Systems(LearningSys) 2015

[19] Anna Rohrbach and Marcus Rohrbach and Ronghang Hu and Trevor Darrell and Bernt Schiele Grounding of Textual Phrases in Images by Reconstruction ECCV2016

[20] Andrej Karpathy and Armand Joulin and Fei-Fei Li Deep Fragment Embeddings for Bidirectional Image-Sentence Mapping NIPS2014

[21] Andrej Karpathy and Fei-Fei Li Deep Visual-Semantic Alignments for Generating Image Descriptions CVPR2015

[22] Lu Cewu and Krishna Ranjay and Bernstein Michael and Fei-Fei Li Visual Relationship Detection with Language Priors ECCV2016

[23] Muraoka M. Maharjan S. Saito M. Yamaguchi K. Okazaki N. Okatani T. Inui K. Recognizing Open-Vocabulary Relations between Objects in Images Pacific Asia Conference on Language Information and Computation 2016

[24] Ronghang Hu and Marcus Rohrbach and Jacob Andreas and Trevor Darrell and Kate Saenko Modeling Relationships in Referential Expressions with Compositional Modular Networks CVPR2017